EE599 Deep Learning- Project Proposal

April 15, 2019

Project Title: landscape image inpainting based on GAN

Project Team: Zhenye Jiang, Yanbang Kan, Maria Mangassarian

Project Summary: In this project we propose to build a GAN which can fill content in a given landscape image with blank that makes the artificial image natural and real. We will collect data from Flickr, Google Image and some open image databases. A successful outcome would be that given an landscape image with some blank area, the network generates contents to fill in these area and makes the artificial result looks natural.

Data Needs and Acquisition Plan: The landscape images are quite massive on Flickr and Google Image, we have download 145K landscape image with size of equal to or smaller than 500x375 with mountains and rivers as our train dataset.

Primary References and Codebase: We propose to build on the approach used in:

- Alec Radford, Luke Metz, Soumith Chintala, “unsupervised representation learning with deep convolutional generative adversarial networks”.
- Jiahui Yu, ZHe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, Luke Metz, Soumith Chintala, “Generative Image Inpainting with Contextual Attention”
- Ugur Demir, Gozde Unal, “Patch-Based Image Inpainting with Generative Adversarial Networks”
- David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, Antonio Torralba, “GAN Dissection: Visualizing and Understanding Generative Adversarial Networks”
- Chaoyue Wang, Chang Xu, Xin Yao, Bolei Zhou,Dacheng Tao, “Evolutionary Generative Adversarial Networks”
- GitHub codebases: Landscape image inpainting Code, https://github.com/tron32213021/ee599-GAN-Project .

Architecture Investigation Plan: We plan to first read and understand above reference. Then, after being familiar with the GAN architecture and techniques, we will try to build some architectures of GAN to achieve our goal.

Estimated Compute Needs: AWS p2 or p3 instance, $100 GPU usage credits. One of papers said that they trained the network with an image set of 8 million and 500k iteration for 2 months using four K80 GPUs. We will aim at a rather small training set and a flexible size of image about 500x375, but still don't know total time to be used.

Team Roles:

- Zhenye Jiang: Data collection, try one of network architecture from papers and train it.
- Yanbang Kan: Do research about contextual attention and apply it into project.
- Maria Mangassarian: Do research about contextual attention, final presentation, slides, and report

Requested Mentor with Rationale: We request Arnab Sanyal to be our team mentor. We have talked with each other and he has mentioned some ideas about how we get start with the project. We have a good idea of what we want to do and have a good starting point from some papers and we are flexible regarding our mentor assignment.

Plan detail:

1. Basic architecture:
   We try to reconstruct the architecture first from paper “Globally and locally Consistent Image Completion”. It contains two parts: Completion Network which is considered as generator in GAN and Discriminator which includes Local Discriminator and Global Discriminator.

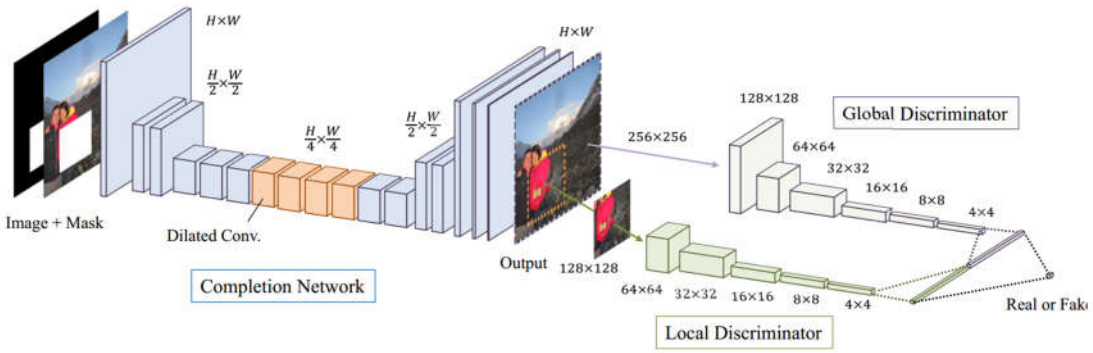

Fig 1: Overview of architecture from “Globally and locally Consistent Image Completion”

Completion Network:

The input of Completion Network is an image with RGB channel and a mask to denote which part of image needs to be inpainted. The size of input is flexible and the completion network is a pixel to pixel architecture using fully convolutional network. The output of completion network will not change pixels out of mask, but only changes the region needs to be inpainted. The convolutional layer with strides equal to 2 is used to down sample the input image rather than the max pooling layer, so that it reduces the memory usage and computational time and extracts features at same time. Dilated convolutional layers are used in the mid-layers. It uses kernel ranging bigger than its input area which allow it to see more information out of input area. This is very important for this task due to it allows network to use pixels far away from inpainting area. Deconvolutional layers are used to restore the image size and complete the pixel to pixel architecture.

Table 1: Architecture of completion network

Type	Kernel	Dilation (η)	Stride	Outputs
conv.	5×5	1	1×1	64
conv.	3×3	1	2×2	128
conv.	3×3	1	1×1	128
conv.	3×3	1	2×2	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
dilated conv.	3×3	2	1×1	256
dilated conv.	3×3	4	1×1	256
dilated conv.	3×3	8	1×1	256
dilated conv.	3×3	16	1×1	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
deconv.	4×4	1	$1/2 \times 1/2$	128
conv.	3×3	1	1×1	128
deconv.	4×4	1	$1/2 \times 1/2$	64
conv.	3×3	1	1×1	32
output	3×3	1	1×1	3

Table 2: Architecture of Discriminator

(a) Local Discriminator

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

(b) Global Discriminator

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

Context Discriminators:

For input images for context discriminators, we should resize them into 256x256 resolutions. Local discriminator accepts an image of size 128x128 including inpainting part for generated image or a same size part of a true image. It is basically to decide whether input image is contextual real in the local part. And global discriminator accepts whole 256x256 image as the input to decide whether the image is contextual real as a whole.

2. Modification:

The initial paper used DC-GAN to train this architecture, but we would like to use WGAN-GP to train the architecture alternatively. Because DCGAN is hard to train and need to adjust parameters from time to time through training and it does not have a criterion to measure the training process. The WGAN-GP, however, has criterion to track. Its loss function denotes the distance between generator's distribution and real images' distribution. It is easily trainable and guaranteed to convergence. The precise details about the loss functions are not set for now, and we will discuss that later.